Session 113

Text on Mac OS X

Xavier Legros
Text & International Technology Manager

Acrobat Reader File Edit Document View Window Help Mon 8:34 PM

Adobe Acrobat Reader ••••

Bookmark

- ヘルプの使い方
  - オンラインヘルプ
  - しおりの使用
  - ナビゲーションパ
  - ヘルプファイルの
  - その他のヘルプリ
- 目次
  - Adobe Acrobat
  - アクセシビリティ
  - Acrobat Reader 5
  - メニューコマンド
  - アクセシビリティ
  - PDF 文書の開き方
  - PDF 文書の表示の
  - 全画面表示で文書
  - PDF 文書の閲覧
  - Web 上での PDF 文
  - PDF 文書の印刷
  - Acrobat Reader
  - PDF 文書情報を見
  - PDF 文書内の検索
  - 注釈のレビュー
  - 注釈の環境設定
  - 電子署名付きの文
  - フォームの入力
  - テキストやグラフ
  - Web Buy の使用
  - UNIX での Acrob
  - 日中韓フォントの
  - オンラインユーザ
  - Windows ショート
  - Macintosh ショー
- 索引
  - A
  - M
  - N
  - P

Adobe Acrobat Reader のヘルプ

ヘルプの使い方 | 目次 | 索引　　戻る ◀ 4 ▶

Adobe Acrobat Reader について

Acrobat Reader を使用すると、Adobe Portable Document Format（PDF）文書を簡単に、表示、閲覧、および印刷することができます。また、フォームに入力したり、ムービーを再生したり、PDF ファイル内を検索することもできます。

Adobe® Acrobat® Reader はアドビシステムズ株式会社の Web サイト（http://

では、Adobe Sites ポップ
す。ただし、サイトに含ま

べて含んでいる必要があり

ることはできません）

- Acrobat Reader エレクトロニックエンドユーザ使用許諾契約書
- Acrobat Reader に含まれる著作権およびその他の所有権情報
- Acrobat Reader を含むすべてのメディアおよびパッケージに以下の情報を提示すること

「Acrobat® Reader Copyright © 1987-2001 Adobe Systems Incorporated. All rights reserved. Adobe、Adobe ロゴ、Adobe Acrobat、および Adobe Acrobat ロゴは、Adobe Systems Incorporated（アドビシステムズ社）の商標です。」

Acrobat Reader エレクトロニックエンドユーザ使用許諾契約書および所有権情報は Acrobat Reader または Acrobat Reader インストーラアプリケーションに含まれています。Acrobat Reader ソフトウェアのインストーラを変更または作成することは固く禁じられています。Acrobat Reader 製品の使用許諾契約書について詳しくは、各製品のインストール中に表示される Acrobat Reader エレクトロニックエンドユーザ使用許諾契約書を参照してください。

Acrobat Readerの配布時に専用の Adobe Acrobat ロゴを使用することができます。詳しく

Find

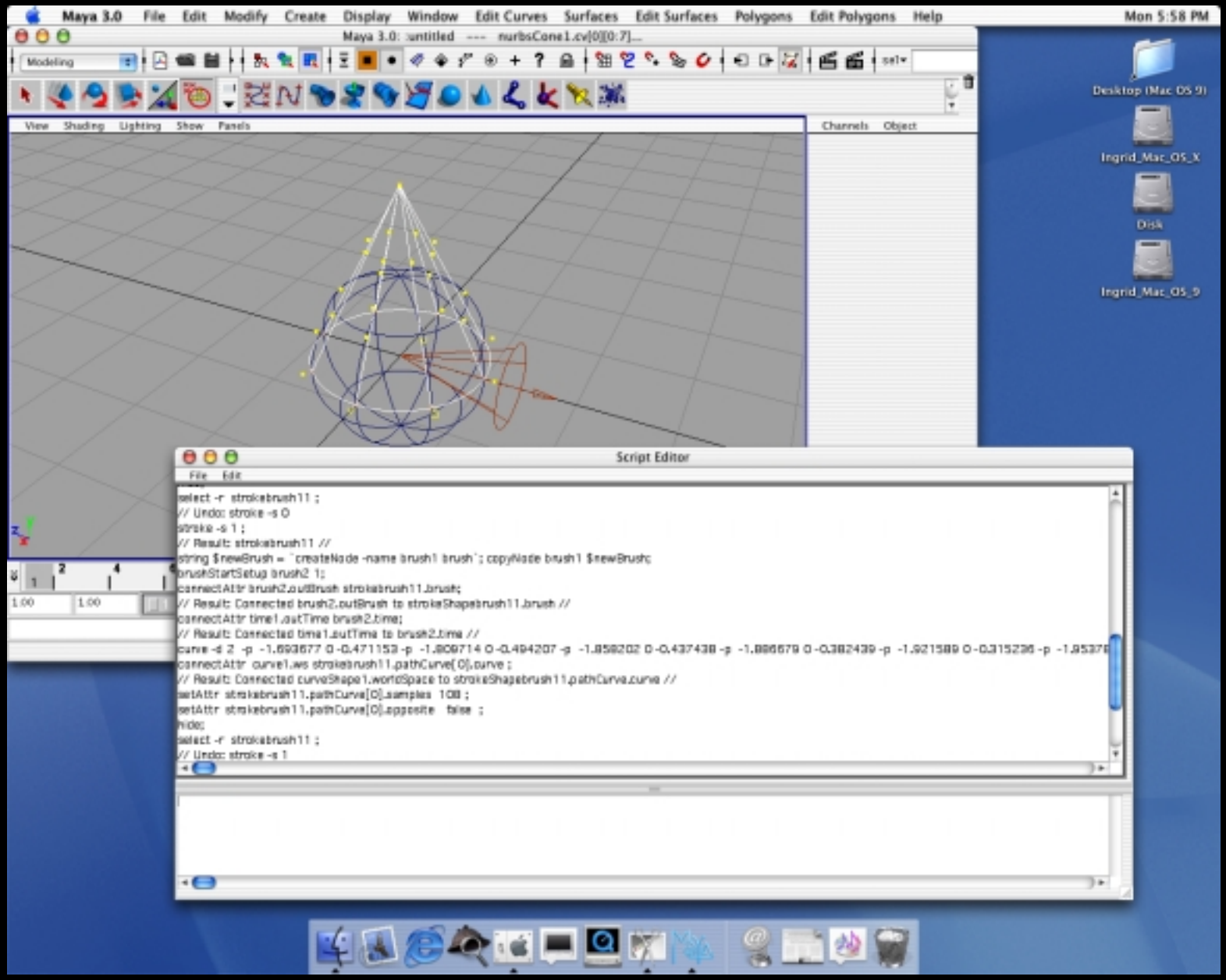
Find What: Adobe Acrobat Reader のヘルプ

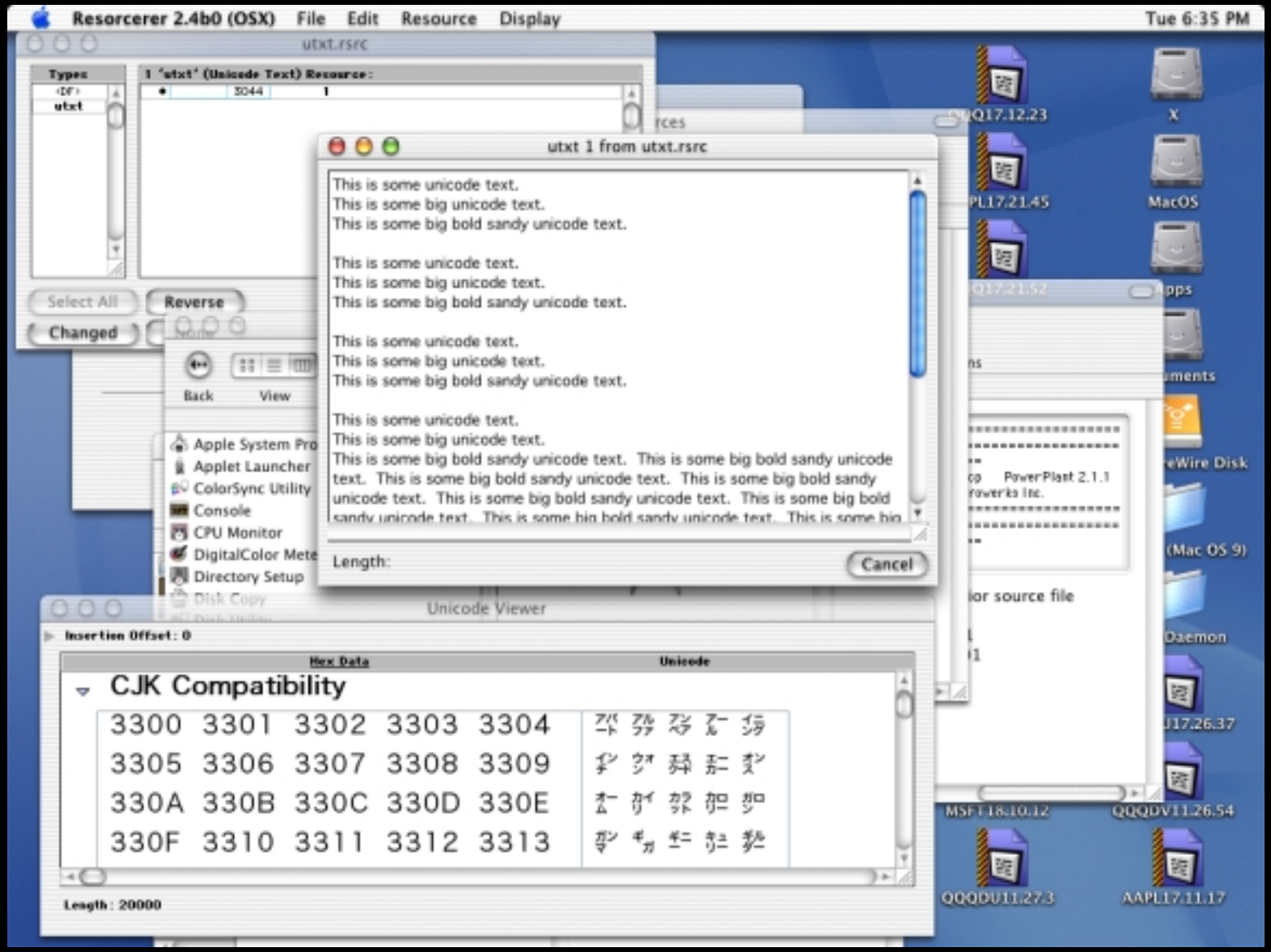
Find　Cancel

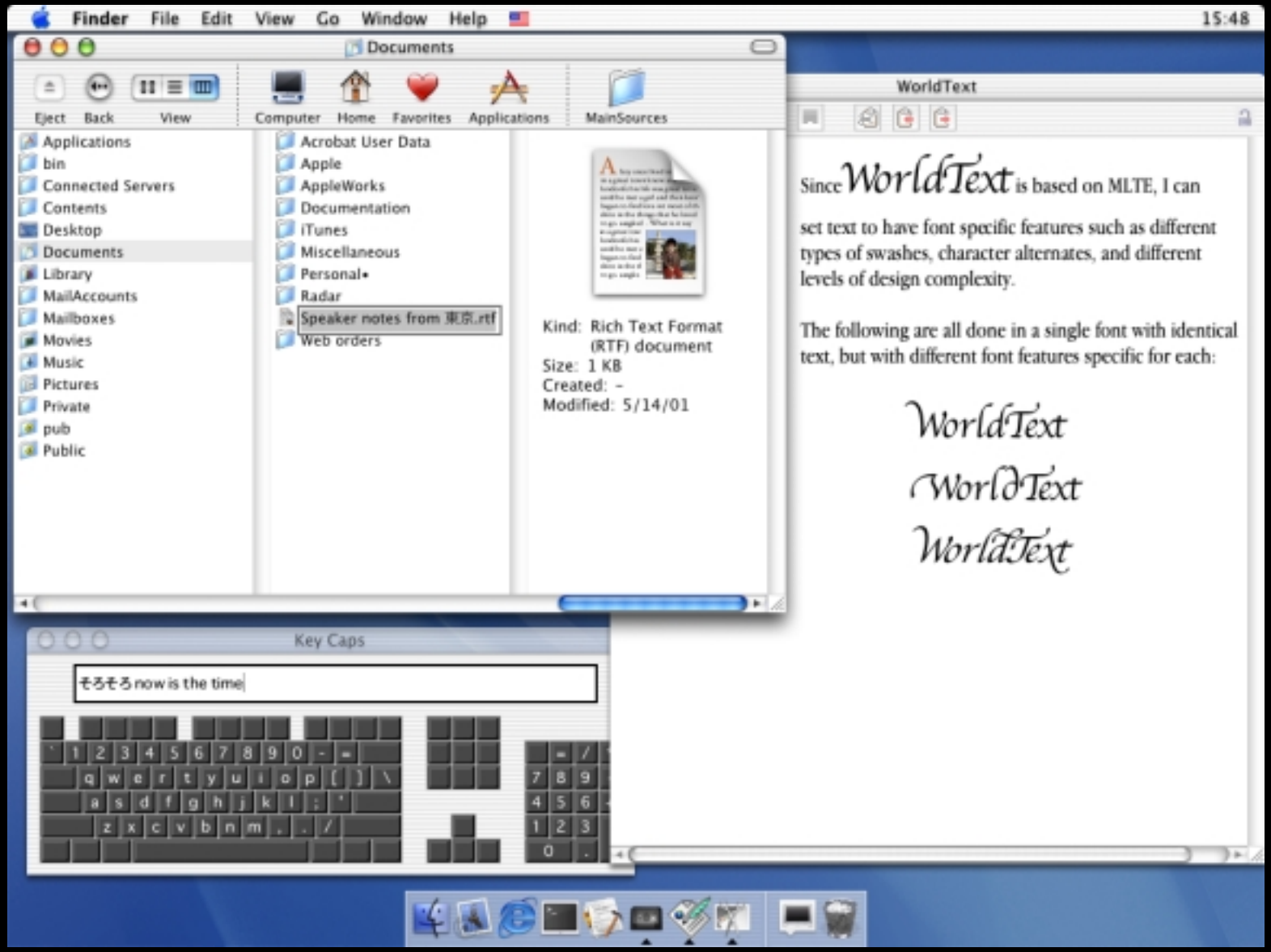
- Match Whole Word Only
- Match Case
- Find Backwards
- Ignore Asian Character Width

Adobe Acrobat Reader 5.0

Trash


Maya 3.0 File Edit Modify Create Display Window Edit Curves Surfaces Edit Surfaces Polygons Edit Polygons Help
Mon 5:58 PM
Maya 3.0: untitled --- nurbsCone1.cv[0][0:7]...
Modeling
View Shading Lighting Show Panels
Channels Object
Desktop (Mac OS 9)
Ingrid_Mac_OS_X
Disk
Ingrid_Mac_OS_9
Script Editor
File Edit
select -r strokebrush11 ;
// Undo: stroke -s 0
stroke -s 1 ;
// Result: strokebrush11 //
string $newBrush = `createNode -name brush1 brush`; copyNode brush1 $newBrush;
brushStartSetup brush2 1;
connectAttr brush2.outBrush strokebrush11.brush;
// Result: Connected brush2.outBrush to strokeShapebrush11.brush //
connectAttr time1.outTime brush2.time;
// Result: Connected time1.outTime to brush2.time //
curve -d 2 -p -1.693677 0 -0.471153 -p -1.809714 0 -0.494207 -p -1.858202 0 -0.437438 -p -1.886679 0 -0.382439 -p -1.921589 0 -0.315236 -p -1.95376
connectAttr curve1.ws strokebrush11.pathCurve[0].curve ;
// Result: Connected curveShape1.worldSpace to strokeShapebrush11.pathCurve.curve //
setAttr strokebrush11.pathCurve[0].samples 100 ;
setAttr strokebrush11.pathCurve[0].opposite false ;
hide;
select -r strokebrush11 ;
// Undo: stroke -s 1

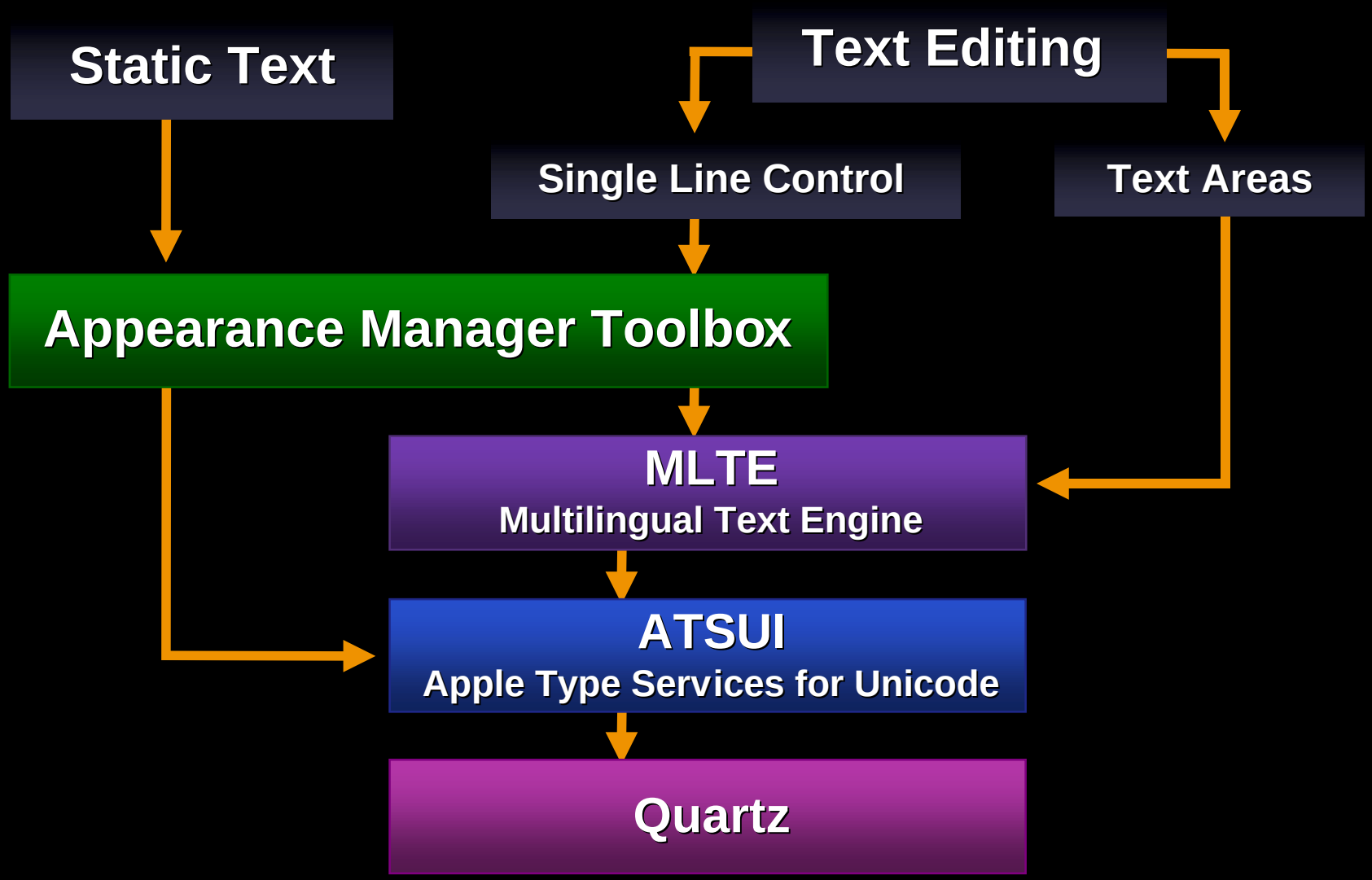

Resorcerer 2.4b0 (OSX)
File
Edit
Resource
Display
Tue 6:35 PM
utxt.rsrc
Types:
utxt
1 'utxt' (Unicode Text) Resource:
Select All
Reverse
Changed
utxt 1 from utxt.rsrc
This is some unicode text.
This is some big unicode text.
This is some big bold sandy unicode text.
Length:
Cancel
Back
View
Apple System Pro
Applet Launcher
ColorSync Utility
Console
CPU Monitor
DigitalColor Mete
Directory Setup
Disk Copy
Unicode Viewer
Insertion Offset: 0
Hex Data
Unicode
CJK Compatibility
3300 3301 3302 3303 3304
3305 3306 3307 3308 3309
330A 330B 330C 330D 330E
330F 3310 3311 3312 3313
Length: 20000
X
MacOS
Apps
Documents
eWire Disk
(Mac OS 9)
Daemon
QQQDV11.26.54
QQQDU11.27.3
AAPL17.11.17


Finder File Edit View Go Window Help
15:48
Documents
Eject Back View
Computer Home Favorites Applications
MainSources
Applications
bin
Connected Servers
Contents
Desktop
Documents
Library
MailAccounts
Mailboxes
Movies
Music
Pictures
Private
pub
Public
Acrobat User Data
Apple
AppleWorks
Documentation
iTunes
Miscellaneous
Personal
Radar
Speaker notes from 東京.rtf
Web orders
Kind: Rich Text Format (RTF) document
Size: 1 KB
Created: –
Modified: 5/14/01
WorldText
Since WorldText is based on MLTE, I can set text to have font specific features such as different types of swashes, character alternates, and different levels of design complexity.
The following are all done in a single font with identical text, but with different font features specific for each:
WorldText
WorldText
WorldText
Key Caps
そろそろ now is the time

Agenda

- Text Architecture
- Displaying static text
- Unicode Edit Text Control
- Text areas with MLTE

. . .and a


Static Text
Text Editing
Single Line Control
Text Areas
Appearance Manager Toolbox
MLTE
Multilingual Text Engine
ATSUI
Apple Type Services for Unicode
Quartz

Solutions for Text

John Harvey
Senior Software Engineer

Text Processing

- Displaying static text
- Unicode Edit Text Control
- Multilingual Text Engine (MLTE)

Displaying Static Text

- Toolbox APIs
  - ThemeText APIs
- ATSUI APIs

ThemeText APIs

- Rendering uses Quartz
  - DrawThemeTextBox()
- Measuring
  - GetThemeTextDimensions()
  - GetThemeTextShadowOutset()
- Truncation
  - TruncateThemeText()
- Easy replacement for DrawText(), TextWidth(), TETextBox(), etc.

ThemeFontIDs

- Standard metafonts
  - System font, Views font, Push Button font, Window Title font, etc.
- Imply font, size, style
- Shadowing effects
- Unicode savvy

ThemeText Performance

- Use the CGContextRef parameter
- Avoid redundant/unnecessary calls
- Cache output values
- We're making things faster…

Text Processing

- Displaying static text
- Unicode Edit Text Control
- Multilingual Text Engine (MLTE)

Unicode Edit Text Control

- Uses Quartz to render text
- Automatic support for CJK input
- Built on top of MLTE
- Creating the control
  - CreateEditUnicodeTextControl (X only)
  - NewControl or GetNewControl
    - procID == kControlEditUnicodeTextProc

Text Processing

- Displaying static text
- Unicode Edit Text Control
- Multilingual Text Engine (MLTE)

What Can MLTE Do?

- Unicode input and imaging
- No more 32K limit!
- Built-in support for:
  - CJK scripts
  - Scrolling, printing, drag/drop
  - Embedded objects, tabs
  - Multiple undo/redo
- Quartz Rendering!

From TextEdit to MLTE

TextEdit	MLTE
Client writes code to scroll and draw scroll bars	MLTE provides built-in support for scrolling and drawing scroll bars
Client writes code to handle clicks	MLTE provides built-in support for clicks
Client writes code to handle printing	MLTE provides built-in support for printing

Lines of Code

	TextEdit	MLTE
Scrollbar handling code	38	0
Click handling code	136	1
Print handling code	95	1
Total lines of code	269	2

PowerPlant Using MLTE

Greg Dow
Metrowerks

PowerPlant LMLTEView Class

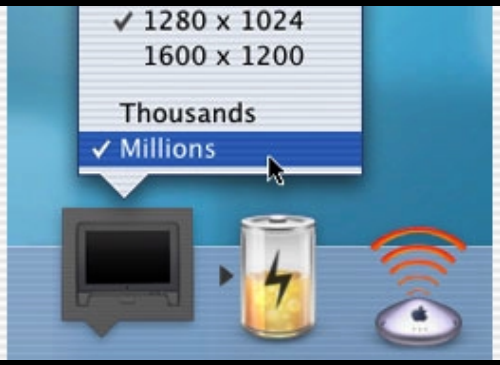
- Uses MLTE
  - More than 32K text
  - Unicode text
  - Embedded pictures, sound, movies
- Replacement for TextEdit Panes
  - Minor changes to update existing code
  - Supports most of old API

PowerPlant LThemeTextBox Class

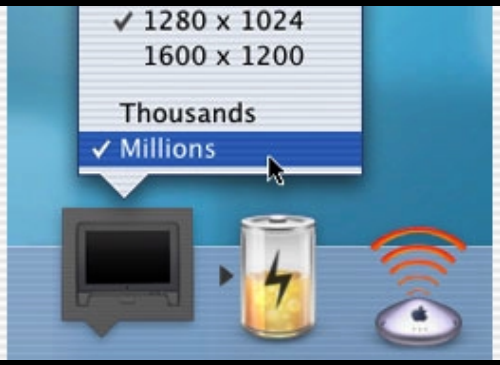
- Anti-aliased text on Mac OS X
- Uses new Toolbox text drawing routines
  - DrawThemeTextBox
  - TXNDrawUnicodeTextBox
  - TXNDrawCFStringTextBox
- More formatting options than StaticText control

Demo

PowerPlant



MLTE on Mac OS X



New Features in Mac OS X

- Quartz rendering
- Text input Carbon events
- Built-in support for new architectures
  - Printing
  - Scrap Manager

Quartz Rendering

- Using Quartz in MLTE provides system style anti-aliasing
- How to:
  - Step 1: Create a CGContext from the port
  - Step 2: Pass the CGContext to MLTE
  - Step 3: There is no step 3!

Requesting Quartz Rendering

```
TXNControlTag           controlTags[] = { kATSUCGContextTag };
TXNControlData          controlData[1];
CGContextRef            cgContext;

if ( CreateCGContextForPort( aGrafPort, &cgContext ) == noErr )
{
   controlData[0].uValue = (UInt32)cgContext;
   status = TXNSetTXNObjectControls(
                        theMLTEObject,
                        false,
                        sizeof( controlTags ) / sizeof( TXNControlTag ),
                        controlTags,
                        controlData );
}
```

New Features in Mac OS X

- Quartz rendering
- Text input Carbon events
- Built-in support for new architectures
  - Printing
  - Scrap Manager

Text Input Carbon Events

- Supports most text input event kinds
  - **kEventTextInputUpdateActiveInputArea**
  - **kEventTextInputUnicodeForKeyEvent**
  - **kEventTextInputOffsetToPos**
  - **kEventTextInputPosToOffset**
  - **kEventTextInputGetSelectedText**
- Supports both pre- and post-process filtering
- Provides an optimal path for text input

Using Text Input Carbon Events

```
// Declare the variables
TXNControlTag          controlTags[] = { kTXNUseCarbonEvents };
TXNControlData         controlData[1];
TXNCarbonEventInfo     carbonEventInfo;
CFStringRef            keys[] = { kTXNTextInputHandlerKey };
EventTargetRef         values[] =
                       { GetWindowEventTarget( myWindowRef ) };
```

Using Text Input Carbon Events

```
// Initialize the TXNCarbonEventInfo data structure
carbonEventInfo.useCarbonEvents = true;
carbonEventInfo.filler = 0;
carbonEventInfo.flags = kTXNNoAppleEventHandlersMask;
carbonEventInfo.fDictionary =
    CFDictionaryCreate( kCFAllocatorDefault,
                        (const void **)&keys,
                        (const void **)&values,
                        1,
                        &kCFCopyStringDictionaryKeyCallBacks,
                        NULL );

controlData[0].uValue = (UInt32)&carbonEventInfo;
```

Using Text Input Carbon Events

```
// Tell MLTE to install its handlers
status = TXNSetTXNObjectControls(
            theMLTEObject,
            false,
            sizeof( controlTags ) / sizeof( TXNControlTag ),
            controlTags,
            controlData );

// Release the dictionary
CFRelease( carbonEventInfo.fDictionary );
```

New Features in Mac OS X

- Quartz rendering
- Text input Carbon events
- Built-in support for new architectures
  - Printing
  - Scrap Manager

Tips and Techniques

Rick Fabrick
Software Engineer

Tips and Techniques

- View and destination rectangles
- Setting style of text range
- Filtering text
- Installing your own drag handlers

Destination Rectangle

View and Destination Rectangles

- Initially set when TXNNewObject is called
- Rect parameter defines the initial value of the rectangles

Rect	View	Destination
NULL	window bounds	US letter size
valid	Rect	Rect minus scrollbars

Changing the Rectangles

- Both view and destination rects
  - TXNResizeFrame
- Only the view rect
  - TXNSetFrameBounds
- Only the destination rect
  - First call TXNResizeFrame
  - Then call TXNSetFrameBounds

View and Destination Options

- To keep view and destination rects in synch
  - kTXNAlwaysWrapAtViewEdgeMask
- To make width of destination rectangle infinite
  - kTXNWordWrapStateTag

Tips and Techniques

- View and destination rectangles
- Setting style of text range
- Filtering text
- Installing your own drag handlers

Setting Style of Text Range

```
// Declare the attributes and their corresponding values
RGBColor            fontColor = { 0xFFFF, 0, 0 };
UInt32              fontSize = 36 << 16;   // needs to be in Fixed format
TXNTypeAttributes   attributes[] = {
   { kTXNQDFontStyleAttribute, kTXNQDFontStyleAttributeSize, bold },
   { kTXNQDFontColorAttribute, kTXNQDFontColorAttributeSize, &fontColor },
   { kTXNQDFontSizeAttribute, kTXNQDFontSizeAttributeSize, fontSize }
};

// Call TXNSetTypeAttributes
status = TXNSetTypeAttributes( myMLTEObject, 3, attributes,
                               kTXNUseCurrentSelection,
                               kTXNUseCurrentSelection );
```

Tips and Techniques

- View and destination rectangles
- Setting style of text range
- Filtering text
- Installing your own drag handlers

Filtering Text

```
// Write a text input Carbon event handler
static pascal OSStatus TextInputHandler(
                                        EventHandlerCallRef         inHandlerRef,
                                        EventRef                    inEvent,
                                        void*                       userData ) {

   if ( GetEventClass( inEvent ) == kEventClassTextInput  &&
        GetEventKind( inEvent ) == kEventTextInputUnicodeForKeyEvent ) {

          if ( EventTextIsNumber( inEvent ) )
                return eventNotHandledErr;       // continue event processing
          else
                return noErr;                    // stops event processing
   }
}
```

Filtering Text With Input Methods

```
// Write a text input Carbon event handler
static pascal OSStatus TextInputHandler(
                                        EventHandlerCallRef         inHandlerRef,
                                        EventRef                    inEvent,
                                        void*                       userData ) {

    if ( GetEventClass( inEvent ) == kEventClassTextInput  &&
         GetEventKind( inEvent ) == kEventTextInputUpdateActiveInputArea ) {

              // After checking for inline session completion
              if ( EventTextIsNumber( inEvent ) )
                   return eventNotHandledErr;       // continue event processing
              else
                   return noErr;                    // stops event processing
    }
}
```

Tips and Techniques

- View and destination rectangles
- Setting style of text range
- Filtering text
- Installing your own drag handlers

Using Your Own Drag Handlers

- Tell MLTE not to install its drag handlers
- Install your drag tracking handler
- Install your drag receive handler

Tell MLTE Not to Install Handler

```
// Create the MLTE object with the “don’t install drag procs” option
status = TXNNewObject( NULL,
                       theWindow,
                       NULL,
                       kTXNDoNotInstallDragProcsMask,
                       kTXNTextEditStyleFrameType,
                       kTXNTextFile,
                       kTXNSystemDefaultEncoding,
                       &theObject,
                       &theFrameID,
                       yourRefCon );
```

Install Your Drag Tracking Handler

```
// Define your DragTrackingHandlerUPP
static pascal OSErr MyDragTrackingProc(
                        DragTrackingMessage     message,
                        WindowPtr               theWindow,
                        void*                   refCon,
                        DragReference           theDragRef ) {

    // Do something if you choose
    DisplaySomeEyeCandy( message );

    // Pass the handling on to MLTE
    return TXNDragTracker( (TXNObject)refCon,
                           GetTheFrameID( theWindow ),
                           message,
                           theWindow,
                           theDragRef, false );
}
```

Install Your Drag Receive Handler

```
// Define your DragReceiveHandlerUPP
static pascal OSErr MyDragReceiveProc(
                                WindowPtr          theWindow,
                                void*              refCon,
                                DragReference      theDragRef ) {

   // Do the filter test
   if  ( DragDataIsANumber( theDragRef ) )
         // If input is accepted, pass the handling on to MLTE
         return TXNDragReceiver( (TXNObject)refCon,
                                 GetTheFrameID( theWindow ),
                                  theWindow,
                                  theDragRef,
                                  false );
   else
         return noErr;     // Stops event processing
}
```

What Should I Use Again…?

Xavier Legros

In Order To…

- Display static text
  - Use HIToolbox APIs

In Order To…

- Display static text
  - Use HIToolbox APIs
- Have editable text areas
  - Use Unicode Edit Text Control

In Order To…

- Display static text
  - Use HIToolbox APIs
- Have editable text areas
  - Use Unicode Edit Text Control
  - Use MLTE API

Additional Information

- MLTE Technical Documentation
  <http://developer.apple.com/techpubs/macosx/Carbon/>
- Bound documentation from Fatbrain
  <http://www.fatbrain.com/>
- Sample Code
  <http://developer.apple.com/techpubs/macosx/Carbon>
- ATSUI Technical Documentation
  <http://developer.apple.com/techpubs/macosx/Carbon/>

Roadmap

122 Using Cocoa	Hall 2 **Thurs., 3:30pm**
128 Font Management on Mac OS X	Room C **Thurs., 2:00pm**
132 Graphics & Imaging Tips & Tricks	Room A2 **Fri., 9:00am**
FF105 Feedback Forum: International Technologies	Room C **Thurs., 5:00pm**

Who to Contact

Xavier Legros
Text & International Technology Manager
Apple Worldwide Developer Relations
xavier@apple.com

Q&A

Xavier Legros
Text & International Technology Manager
Apple Worldwide Developer Relations
xavier@apple.com

http://developer.apple.com/events/urls.html

WWDC

Worldwide Developers Conference 2001